

**THE CLAIMS**

1. (Previously Amended) A system for managing allocation of computer resources among applications operating on a computer, based upon at least one user-defined goal for at least one application and application performance information related to the at least one application, the system comprising:

a calculator that determines an allocation request value using a proportional factor, an integral factor, and a derivative factor, wherein the factors are calculated from the at least one user-defined goal and the performance information; and

an arbiter that is operative to adjust the allocation request value into an adjusted allocation value when the allocation request value and allocation request values associated with other applications exceeds a predetermined value.

2. (Previously Amended) The system of claim 1, wherein:

the proportional factor includes a proportional constant and a proportional variable, wherein the proportional constant is a user-selectable input, and the proportional variable is determined from the at least one user-defined goal and the performance information;

the integral factor includes a integral constant and a integral variable,

wherein the integral constant is a user-selectable input, and the integral variable is determined from the at least one user-defined goal and the performance information; and

the derivative factor includes a derivative constant and a derivative variable,

wherein the derivative constant is a user-selectable input, and the derivative variable is determined from the at least one user-defined goal and the performance information.

3. (Previously Amended) The system of claim 2, wherein:

the proportional factor is  $(KP \times P_i^*)$ , wherein  $KP$  is the proportional constant and  $P_i^*$  is the proportional variable, and  $P_i^* = P_i - V_i$ , wherein  $P_i$  is performance information and  $V_i$  is the at least one user-defined goal;

the integral factor is  $(KI \times I_i)$ , wherein  $KI$  is the integral constant and  $I_i$  is the integral variable, and  $I_i = (I_{iold} \times IH_i) + P_i^*$ ,  $I_{iold}$  is a previous value for  $I_i$ , and  $IH_i$  is an integral history constant which is a user-selectable input, and  $P_i^*$  is the proportional variable; and

the derivative factor is  $(KD \times D_i)$ , wherein  $KD$  is the derivative constant and  $D_i$  is the derivative variable, and  $D_i = (P_i^* - P_i^{*old})/t$ , wherein  $P_i^*$  is the proportional variable,  $P_i^{*old}$  is a previous value for  $P_i^*$ , and  $t$  is a time between the performance information used to calculate  $P_i^*$  and previous performance information used to calculate  $P_i^{*old}$ .

4. (Original) The system of claim 3, wherein:

the allocation request value,  $R_i$ , is determined by  $R_i = (KP \times P_i^*) + (KD \times D_i) + (KI \times I_i) + R_{iold}$ , wherein  $R_{iold}$  is a previous value for  $R_i$ .

5. (Previously Amended) The system of claim 1, wherein a number format for each of the at least one user-defined goal, the performance information, and the allocation request value is selected from the group consisting of:

a floating point number, and an integer number.

6. (Original) The system of claim 1, wherein the allocation request value is a floating point number, the system further comprising:

a rounder that uses cumulative rounding to adjust the floating point allocation request value into an integer number.

7. (Previously Amended) The system of claim 1, wherein:

the performance information is generated by a performance monitor that monitors a characteristic of the application associated with the at least one user-defined goal.

8. (Original) The system of claim 1, wherein:

the allocation request value is used by a process resource manager to allocate computer processing resources among the applications operating on the computer.

9. (Previously Amended) The system of claim 1, wherein the at least one application is one of a plurality of applications, and each application of the plurality has an associated user-selectable priority and an allocation request value, wherein:

the arbiter determines whether each application of the plurality of applications having the same priority can be allocated resources to equal its associated allocation request value, if so, then the arbiter forms the adjusted allocation request value for each application by equaling the adjusted allocation request value to the allocation request value, and if not, then the arbiter determines whether each application of the plurality of applications having the same priority can be allocated resources to equal a target value.

10. (Original) The system of claim 9, wherein:

the target value is selected by the arbiter from the lowest of a previously allocated request value, which has not been previously selected as a target value, and an allocation request value of an application of the plurality of applications having the same priority, which has not been previously selected as a target value.

11. (Original) The system of claim 9, wherein:

the arbiter forms the adjusted allocation request value for each application by equaling the adjusted allocation request value to the target value, if the arbiter determines that each application of the plurality of applications having the same priority can be allocated resources to equal the target value.

12. (Previously Amended) A computer program product having a computer readable medium having computer program logic recorded thereon for managing allocation of computer resources among applications operating on a computer, based upon at least one user-defined goal for at least one application and application performance information related to the at least one application, the computer program product comprising:

code for determining an allocation request value using a proportional factor, an integral factor, and a derivative factor, wherein the factors are calculated from the at least one user-defined goal and the performance information; and

code for adjusting the allocation request value, and is operative when the allocation request value and allocation request values associated with other applications exceeds a predetermined value.

13. (Previously Amended) The computer program product of claim 12, wherein:  
the proportional factor includes a proportional constant and a proportional variable,  
wherein the proportional constant is a user-selectable input, and the proportional variable is  
determined from the at least one user-defined goal and the performance information;  
the integral factor includes a integral constant and a integral variable,  
wherein the integral constant is a user-selectable input, and the integral variable is  
determined from the at least one user-defined goal and the performance information; and  
the derivative factor includes a derivative constant and a derivative variable,  
wherein the derivative constant is a user-selectable input, and the derivative variable  
is determined from the at least one user-defined goal and the performance information.

14. (Previously Amended) The computer program product of claim 13, wherein:  
the proportional factor is  $(KP \times P_i^*)$ , wherein  $KP$  is the proportional constant and  $P_i^*$   
is the proportional variable, and  $P_i^* = P_i - V_i$ , wherein  $P_i$  is performance information and  $V_i$   
is the at least one user-defined goal;  
the integral factor is  $(KI \times I_i)$ , wherein  $KI$  is the integral constant and  $I_i$  is the integral  
variable, and  $I_i = (I_{iold} \times IH_i) + P_i^*$ ,  $I_{iold}$  is a previous value for  $I_i$ , and  $IH_i$  is an integral  
history constant which is a user-selectable input, and  $P_i^*$  is the proportional variable; and  
the derivative factor is  $(KD \times D_i)$ , wherein  $KD$  is the derivative constant and  $D_i$  is the  
derivative variable, and  $D_i = (P_i^* - P_i^{*old})/t$ , wherein  $P_i^*$  is the proportional variable,  $P_i^{*old}$   
is a previous value for  $P_i^*$ , and  $t$  is a time between the performance information used to  
calculate  $P_i^*$  and previous performance information used to calculate  $P_i^{*old}$ .

15. (Original) The computer program product of claim 14, wherein:  
the allocation request value,  $R_i$ , is determined by  $R_i = (KP \times P_i^*) + (KD \times D_i) + (KI \times I_i) + R_{iold}$ , wherein  $R_{iold}$  is a previous value for  $R_i$ .

16. (Previously Amended) The computer program product of claim 12, wherein the at least one application is one of a plurality of applications, and each application of the plurality has an associated user-selectable priority, wherein the code for adjusting the allocation request value comprises:

code for determining whether each application of the plurality of applications having the same priority can be allocated resources to equal its associated allocation request value, if so, then the adjusted allocation request value for each application is formed by equaling the adjusted allocation request value to the allocation request value, and if not, then whether each application of the plurality of applications having the same priority can be allocated resources to equal a target value is determined;

wherein the target value is selected from the lowest of a previously allocated request value, which has not been previously selected as a target value, and an allocation request value of an application of the plurality of applications having the same priority, which has not been previously selected as a target value.

17. (Original) The computer program product of claim 16, wherein the code for adjusting the allocation request value further comprises:

code for forming the adjusted allocation request value for each application by equaling the adjusted allocation request value to the target value, if each application of the plurality of applications having the same priority can be allocated resources to equal the target value.

18. (Previously Amended) A method for managing allocation of computer resources among applications operating on a computer, based upon at least one user-defined goal for at least one application and application performance information related to the at least one application, the method comprising the steps of:

determining an allocation request value using a proportional factor, an integral factor, and a derivative factor, wherein the factors are calculated from the at least one user-defined goal and the performance information; and

adjusting the allocation request value, when the allocation request value and allocation request values associated with other applications exceeds a predetermined value.

19. (Previously Amended) The method of claim 18, wherein the at least one application is one of a plurality of applications, and each application of the plurality has an associated user-selectable priority, wherein the step of adjusting the allocation request value comprises the step of:

determining whether each application of the plurality of applications having the same priority can be allocated resources to equal its associated allocation request value, if so, then the adjusted allocation request value for each application is formed by equaling the adjusted allocation request value to the allocation request value, and if not, then whether each application of the plurality of applications having the same priority can be allocated resources to equal a target value is determined,

wherein the target value is selected from the lowest of a previously allocated request value, which has not been previously selected as a target value, and an allocation request value of an application of the plurality of applications having the same priority, which has not been previously selected as a target value.

20. (Original) The method of claim 19, wherein the step of adjusting the allocation request value further comprises the step of:

forming the adjusted allocation request value for each application by equaling the adjusted allocation request value to the target value, if each application of the plurality of applications having the same priority can be allocated resources to equal the target value.